



**University of
Zurich^{UZH}**

**Zurich Open Repository and
Archive**

University of Zurich
University Library
Strickhofstrasse 39
CH-8057 Zurich
www.zora.uzh.ch

Year: 2018

Financing the Web of Data with Delayed-Answer Auctions

Grubenmann, Tobias ; Bernstein, Abraham ; Moor, Dmitrii ; Seuken, Sven

Abstract: The World Wide Web is a massive network of interlinked documents. One of the reasons the World Wide Web is so successful is the fact that most content is available free of any charge. Inspired by the success of the World Wide Web, the Web of Data applies the same strategy of interlinking to data. To this point, most of data in the Web of Data is also free of charge. The fact that the data is freely available raises the question of financing these services, however. As we will discuss in this paper, advertisement and donations cannot easily be applied to this new setting. To create incentives to subsidize data providers, we propose that sponsors should pay the providers to promote sponsored data. In return, sponsored data will be privileged over non-sponsored data. Since it is not possible to enforce a certain ordering on the data the user will receive, we propose to split up the data into different batches and deliver these batches with different delays. In this way, we can privilege sponsored data without withholding any non-sponsored data from the user. In this paper, we introduce a new concept of a delayed-answer auction, where sponsors can pay to prioritize their data. We introduce a new model which captures the particular situation when a user access data in the Web of Data. We show how the weighted Vickrey-Clarke-Groves auction mechanism can be applied to our scenario and we discuss how certain parameters can influence the nature of our auction. With our new concept, we build a first step to a free yet financial sustainable Web of Data.

DOI: <https://doi.org/10.1145/3178876.3186002>

Posted at the Zurich Open Repository and Archive, University of Zurich

ZORA URL: <https://doi.org/10.5167/uzh-149689>

Conference or Workshop Item

Published Version

Originally published at:

Grubenmann, Tobias; Bernstein, Abraham; Moor, Dmitrii; Seuken, Sven (2018). Financing the Web of Data with Delayed-Answer Auctions. In: WWW 2018: The 2018 Web Conference, Lyon, 23 April 2018 - 27 April 2018, International World Wide Web Conference Committee.

DOI: <https://doi.org/10.1145/3178876.3186002>

Financing the Web of Data with Delayed-Answer Auctions

Tobias Grubenmann
University of Zurich
Department of Informatics
Zurich, Switzerland
grubenmann@ifi.uzh.ch

Dmitry Moor
University of Zurich
Department of Informatics
Zurich, Switzerland
dmoor@ifi.uzh.ch

Abraham Bernstein
University of Zurich
Department of Informatics
Zurich, Switzerland
bernstein@ifi.uzh.ch

Sven Seuken
University of Zurich
Department of Informatics
Zurich, Switzerland
seuken@ifi.uzh.ch

ABSTRACT

The World Wide Web is a massive network of interlinked documents. One of the reasons the World Wide Web is so successful is the fact that most content is available free of any charge. Inspired by the success of the World Wide Web, the Web of Data applies the same strategy of interlinking to data. To this point, most of data in the Web of Data is also free of charge. The fact that the data is freely available raises the question of financing these services, however. As we will discuss in this paper, advertisement and donations cannot easily be applied to this new setting.

To create incentives to subsidize data providers, we propose that sponsors should pay the providers to promote sponsored data. In return, sponsored data will be privileged over non-sponsored data. Since it is not possible to enforce a certain ordering on the data the user will receive, we propose to split up the data into different batches and deliver these batches with different delays. In this way, we can privilege sponsored data without withholding any non-sponsored data from the user.

In this paper, we introduce a new concept of a delayed-answer auction, where sponsors can pay to prioritize their data. We introduce a new model which captures the particular situation when a user access data in the Web of Data. We show how the weighted Vickrey-Clarke-Groves auction mechanism can be applied to our scenario and we discuss how certain parameters can influence the nature of our auction. With our new concept, we build a first step to a free yet financial sustainable Web of Data.

CCS CONCEPTS

• **Information systems** → **Sponsored search advertising**;
Content match advertising; **Online auctions**; • **Applied computing** → **Online auctions**;

KEYWORDS

Web of Data, Slot Auctions, Delay, Sponsored Search Results, Vickrey-Clarke-Groves mechanism

ACM Reference Format:

Tobias Grubenmann, Abraham Bernstein, Dmitry Moor, and Sven Seuken. 2018. Financing the Web of Data with Delayed-Answer Auctions. In *WWW 2018: The 2018 Web Conference, April 23–27, 2018, Lyon, France*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3178876.3186002>

1 INTRODUCTION

The Web of Data (WoD) is the result of applying the principle of interlinking documents, which fueled the growth of the World Wide Web, to data, which results in so called *Linked Data*. Like in its predecessor, the WWW, most content of the WoD is to date free of any charge. Allowing users to access Linked Data for free introduces new challenges when it comes to financing these services. The question we are answering in this paper is: *How can the WoD be free and financially sustainable at the same time?*

The World Wide Web is not only the inspiration of the Web of Data but also serves as an example of how to finance such services. Hence, it is natural to think about applying the techniques which finance the WWW to the WoD. One of the biggest financial motors of the WWW is advertisement. What made advertisement in the WWW more efficient than in other medias is the fact that advertisement in the WWW is customized to the user. Hence, it is easier for advertisers to target a specific user group. This customization is mainly achieved either by showing an ad based on a keyword entered in a search page or by embedding ads which are related to a certain webpage's content. Unfortunately, these techniques do not apply that easily to the WoD. The main problem of the WoD with respect to advertisement is that the data provider has no influence on how the content is presented to the user: In the WWW, advertisement can be added to the presentation of the content at the discretion of the data provider. In the WoD, however, a user receives the data in a structured format. This structured format allows algorithms to automatically process the data further, if needed, and does not provide the means to add any additional advertisements. Accessing the WoD from a user's perspective is more similar to accessing a database than accessing a website's content.

Donations could be an alternative to financing the WoD. Wikipedia is an example of a content provider in the WWW which

This paper is published under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW 2018, April 23–27, 2018, Lyon, France

© 2018 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC BY 4.0 License.

ACM ISBN 978-1-4503-5639-8/18/04.

<https://doi.org/10.1145/3178876.3186002>

is financed by donations. However, a significant part of the users is not aware that there is a possibility to donate to Wikipedia or do not know how to donate [13]. Such a lack of *awareness* for the need of donations will be even more pronounced in the WoD. Users in the WoD often access a lot of datasets at the same time. Sometimes these datasets are part of a federation and can be accessed transparently through a single accessing point. Very often, a user might simply not be aware of using a certain dataset let alone realizing that the provider requires financial support through donations.

To remedy this situation, we propose that *sponsors* who are interested in promoting certain data will subsidize *all data provider which are involved in creating the specific solution containing the subsidized data*. A sponsor can be anybody who gains an advantage if certain data is distributed to as much users as possible. However, this means that we need a way to *privilege* highly sponsored over less sponsored or non-sponsored data. In the WWW, search engines, for example, can change the ordering of search results to privilege certain websites and order advertisements on the search result page based on the payments. In the WoD, the query language SPARQL [17] which is used to query the desired data allows a user to specify the ordering of the received data. Even if we disable such a functionality, the structured format in which the data is delivered easily allows a user, or the program which is querying on behalf of the user, to reorder the received data ad libitum. This means that it is not possible to force a certain *ranking* of the received data upon the user within a single query answer. One alternative is to simply deliver only part of the available data and withhold the rest. Data would then be delivered or not based on the amount of sponsorship the data received. This would create a situation where different sponsors compete to be part of the delivered data. Whereas such a situation would create enough competition between sponsors to ensure a certain revenue, it opposes the idea that *all* data should be freely available. Hence, we propose a new concept where the requested data is *delayed* depending on how much a certain sponsor is willing to pay for its data being privileged over the data of other sponsors. Our new concept is aligned with the idea that all data is freely available but creates at the same time an incentive for sponsors to promote certain data. The revenue generated by our concepts can be used to finance the providers which are responsible for hosting and maintaining the data.

By delaying part of the data requested by a user, we are harming the user's experience to a certain degree. Obviously, a user would prefer to get all requested data immediately instead of receiving the data in consecutively delayed batches. We discuss the user's experience in detail in Section 6.3. In addition, a user would prefer to receive data in an unbiased way instead of receiving first the data which received the highest sponsoring. To remedy this problem, we introduce an extension to our model in Section 7 where part of the data is randomly assigned to batches.

The contributions of our paper are: (1) we introduce our new concept of a *delayed-answer auction* where sponsors can pay to have sponsored data delivered quicker to the user, (2) we introduce a new link selection model (akin to click models in the WWW) which captures the probability that a user selects a link contained in a certain solution, (3) we show how the weighted VCG auction mechanism can be applied to our scenario and we discuss how certain parameters of the auction can influence the nature of our

auction, and (4) we discuss an extension to our model where only part of the data delivery is influenced by the sponsors.

2 MOTIVATION

Imagine a user who wants to make a reservation for a restaurant in Zurich, having a rating of at least 8.0, and which is offering traditional food. Such a user can use the query language SPARQL [17] to express the exact needs. Compared to a keyword based searched in the WWW, such a *semantic search* is much more precise when it comes to expressing which data the user actually wants. Listing 1 shows how a SPARQL query for our example could look like.

Listing 1: A query which asks for traditional restaurants in Zurich with a ranking of at least 8.0.

```
SELECT ?name ?link
WHERE {
    ?restaurant :City ex:Zurich .
    ?restaurant :Food_Style :Traditional .
    ?restaurant :Reservation_Link ?link .
    ?restaurant :ranking ?rank . FILTER(?rank >= 8) }
```

The user issuing the SPARQL query from Listing 1 has an incentive to make a reservation for a restaurant with the desired properties as quick and straightforward as possible. Meanwhile, the different restaurants have an incentive that the user makes the reservation at their own restaurant and not at the competition. Assume now that the different solutions for the query arrive at the user with different delays, e.g., the first solution will appear immediately, the second solution after a few seconds, the third solution even a few seconds later, and so on. It is not hard to imagine that the user would not wait an hour for all thousands of solutions to appear. Instead, a user would probably only wait a few seconds before picking one of the already available solutions and make the reservation. Indeed, as [20] implied, users become very impatient over time and even a delay of 250ms can influence a user to visit a website of a close competitor. Similar, [16] argue how important latency is for the user's experience of a website. An experiment at Amazon showed that a delay of 100ms resulted in sale decrease of 1% [18]. Also, an experiment at Google showed that if the time to display search results is increased by 500ms, the revenue is reduced by 20% [18]. Finally, experiments at Microsoft Live Search showed that when the research result page is slowed down by a second, ad clicks per user decline by 1.5%, and even by 4.4% when the delay is two seconds [18].

Based on these findings, we developed a model for delaying solutions for SPARQL query answers where *sponsors* can pay money to prioritize certain solutions and delay others. Using this strategy of delaying part of the query answer, we manage to discriminate different solutions. Similar to a user clicking on a link in the WWW, a user (or program acting on behalf of a user) can decide to look up a certain URI contained in a solution for a query answer. A sponsor is motivated to pay money whenever such a URI lookup directs the user to some service. In our example, the offered service could be a reservation system for the respective restaurant. We will denote URIs which direct a user to some service as *service link*, or just *link*, to distinguish it from other URIs. Depending on how much

a sponsor is willing to pay for a visited service link, a solution containing that link might be more or less prioritized. If a user looks up a service link, all the data provider involved in creating the specific solution will receive a share of the money the sponsor is paying. In particular, the provider offering the rankings of different restaurants gets paid for its service.

In Section 4 we discuss our concept of delaying solutions in more detail and in Section 5 we introduce a formal model for our concept. Note that our model is targeted at query answers for which a delay influences the likelihood that the user will consider a certain solution. This is not always the case. For some queries, a user does not mind waiting a long time receiving an answer. Hence, it is important to keep in mind that our concept is specifically designed for situations where such delays do matter.

3 RELATED WORK

Auctions for Sponsored Search Results: Before Google introduced the Generalized Second Price (GSP) auction in 2002, first price models were used for selling ads on search result pages [27]. The idea of the GSP auction is that each bidder submits one bid, which indicates the bidders value for a click. Different positions (or slots) on the search result page will have different click-through rates. Hence, the bidders value per click translates into a value per slot. Depending on the auctioneer, different ads might have different click-through rates for the same slot, adding an additional layer of complexity. In the simplest variation of the GSP auction, the first slot is given to the bidder declaring the highest value, the second slot to the bidder with the next highest value, and so on. What gives the name to the GSP auction is the fact that each bidder pays a price which equals the value of the next lowest bidder, also known as the *second price*. [11]

The advantage of the GSP auction over first price models is that the GSP auction prevents “cycling” patterns, a situation where prices gradually rise until a sudden drop occurs and the pattern starts over [10]. Shortly after Google implemented the GSP auction, one of their engineers realized that a Vickrey-Clarke-Groves (VCG) auction [7, 14, 26] could also be implemented to sell ad positions on their search results pages. The advantage of the VCG auction is that it is truthful, which means that each bidder has an incentive to report their true value as bid. Google refrained from replacing the GSP with the VCG auction because the GSP auction was already growing attention and there would have been additional effort involved, both on the side of Google and the sellers, to change to the VCG auction. [25]

In [1], Aggarwal et al. propose a position-based auctions mechanism where bidders can impose additional constraints on the positions they want their ad to appear. The motivation behind this auction format is that a bidder might have a value if the ad just appears at the top of a search result page, even if the user does not click the ad, because the mere appearance of the ad increases the visibility of the brand.

In the WoD setting, it is unclear which of these auctions are applicable. In section 5.3 we will show how the VCG auction mechanism can be applied to our setting.

Click Models for Web Search and Sponsored Search Auctions: Both web search and sponsored search auctions need click

models for estimating how likely a user would select a certain search result or a certain ad. Hence, the different models used in these two fields are often overlapping and can often be used in the other field.

The simplest models for user clicks on advertisements is based on the assumption that click-through rates can be separated into two factors: one factor which is only influenced by the ad which wins the slot and one factor which is only influenced by the slot position itself. The result of this assumption, called the *separability assumption*, is that the click-through rate of a certain ad winning a certain slot can be computed by just multiplying these two factors, without having to consider which ads are winning all the other slots. In settings where the click-through rates are not separable, a weighted VCG mechanism may not apply. [3]

The click model of [23] provides one example of such separable factors: the influence of the slot is described as the probability that a user sees an ad at a specific position. The influence of the ad is described as the probability that the user clicks on the ad given that the user saw the ad. Note that it is assumed that those probabilities are independent of the ads already shown to the user.

Aggarwal et al. [2] introduce a Markovian user model where the user scans a list of ads and makes a decision whether to (1) click on the ad, (2) continue scanning the list, or (3) abort the inspections of the ads. Note that in this model, the probability that a user clicks on a certain ad does not only depend on the ad itself and the position but also on the ads placed in higher slots. Hence, the separability assumption does not hold anymore in this model. A consequence of this is that the GSP allocation of ads is not anymore the most efficient allocation [2]. Similar to the model of [2], [8] introduces a cascading model for search results where each document is either clicked or skipped. In the latter case, the user continues the scanning of the list of results.

One drawback of the models of [2] and [8] is that it is assumed that a user will only continue scanning the ads/search results if no previous ad/search result has been clicked, yet, and hence, assuming that the user will click at most on one ad/search result during the scanning process. [15] introduced the dependent click model which extends the cascading model by introducing conditional probabilities of a user to continue scanning the list of results depending on whether a click occurred on the current document or not and hence, allowing for multiple clicks within a single scanning process.

In [28], Zhu et al. introduce their *General Click Model* which is a more general model for user clicks. Most existing models can be considered a special case of their general model and they showed, for example, that the models of [23] and [8] can be modelled using their General Click Model.

As described in Section 2, a user (or program) can visit certain service links, which are URIs which direct the user to some service. Selecting such a service link is very similar to a user clicking on a link in the WWW. However, despite the variety of click models for web search and sponsored search auctions, none of them captures the situation when a user selects a service link from a set of solutions for a SPARQL query. Hence, in Section 5.1 we introduce a new selection model which we designed especially for SPARQL query answers.

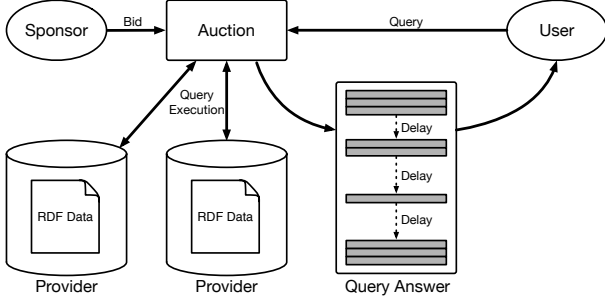


Figure 1: A user gets a delayed query answer based on the bids of sponsors.

4 DELAYED-ANSWER AUCTION

In this section, we introduce our concept of a *delayed-answer auction*. At the core of our concept lies the ability of *sponsors* to pay money if a user follows a certain service link contained in a query solution. The data is accessible in form of a SPARQL endpoint. Our auction mechanism makes sure that solutions containing links with higher bids appear with a smaller delay than solutions containing links with lower bids. Hence, our auction mechanism creates a ranking of the solutions by introducing different delays for them.

The user who poses the SPARQL query and the data providers which offer the data needed to answer the query define the context of our auction. Given this context, the sponsors are the participants (or bidder) in the auction. Hence, we will use the expressions *sponsor* and *bidder* interchangeably. The bidders place a *bid* on a specific link contained in a query solution. This bid indicates how much the sponsor is willing to pay if the user visits a certain service link. We call a service link which has a bid placed on it a *sponsored link*.

In our concept, we make the simplifying assumption that each solution contains at most one sponsored link. This means that each solution can be associated with one sponsor and one bid. If nobody is bidding on a link contained in a solution or the solution contains no link at all, the bid is set to 0. We discuss the more general case when there might be multiple sponsored links in a single solution in the limitations in Section 8. If a user visits a link contained in a solution for a query answer, the visit of the service link is registered at the auctioneer and the placed bid is charged to the respective bidder. The revenue generated by the auction is used to finance all the entities involved in providing the data which were needed to create the solutions for the query answer.

Figure 1 illustrates the process of our delayed-answer auction concept. A user submits a query to the auction. The auction executes the query and generates the query answer. Parts of the query answer get delayed, depending on how much the sponsors bid on certain links contained in the solutions. Figure 2 illustrates how the money gets redistributed by the auction to the providers which were involved in creating the query answer. The service link which is visited directs the user to the auction which, in turn, redirects the user to the appropriate location. This redirecting mechanism is important to keep track of which links are visited.

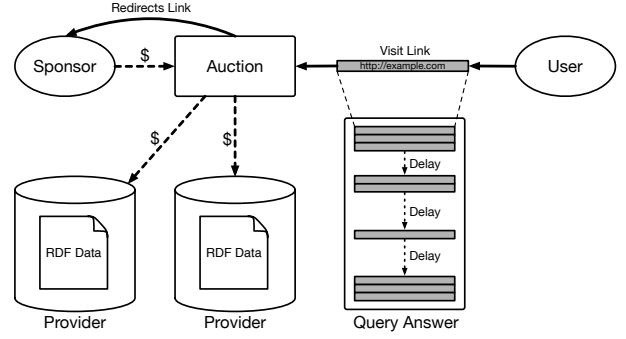


Figure 2: If a user visits a link, the sponsor of that link pays the auction which distributes the money among the providers.

To achieve the link redirection functionality, the auction has to replace each occurrence of a sponsored link with a new link that directs the user to the auction when visited. The auction has to track which link replacement corresponds to which original link and redirect the user (or program), accordingly. The revenue generated by the auction is used to finance the provider of the data. Not only the provider providing the sponsored link but all providers providing any relevant data for the solution which contained the visited link gets a share of the revenue. In addition, it is possible to use part of the revenue to subsidize providers which were not so lucky and could not (or not enough) contribute to different sponsored links to cover the operation costs of their services. The exact distribution of the revenue among the providers is beyond the scope of this paper, however. It is important to note that for the auction to work properly, the providers and the auction need to build a closed system, meaning that the providers' data are only accessible through the auction itself and cannot be accessed directly by the user. If the providers and the auction would not build a closed system, a user could circumvent the auction mechanism and directly query the provider's data without suffering from the delays introduced by the auction.

Note that delaying certain solutions is not part of the standard SPARQL protocol [12]. Instead, the query answer must be offered as a *stream* of solutions. There are various extensions of SPARQL which offer the possibility to return solutions in form of a stream [4–6, 9, 19, 22, 24]. A detailed discussion of the advantages or disadvantages of the different streaming solutions is beyond the scope of this paper, however.

5 FORMAL MODEL

In this section, we introduce the formal model of our concept. First, we introduce our new link selection model and then, we discuss the how this model can be applied to a weighted VCG auction.

5.1 Batch Link Model

Our *Batch Link Model* differs from traditional models for sponsored search results. In click models for sponsored search results an ad is shown to a user if a bidder bids on a certain *keyword*. This relevance

of different ads for a specific keyword can differ quite a lot because (1) the keyword entered by a user can be ambiguous or unspecific because of the lack of any semantics and (2) an advertiser can decide to bid for a certain keyword, even if the ad is not very relevant. In a semantic search, the ambiguity is mostly removed and, given that the user, or a program acting on behalf of the user, formulated the query diligently enough, the lack of specificity as well. In addition, a bidder cannot force a certain service link to be part of the query answer by increasing the bid enough. Instead, only links which actually match the query are allowed to be in the query answer. Hence, in our model, we assume that each service link i contained in a solution for a query answer has the *same* probability p_{rel} of being relevant for the query. The probability p_{rel} depends on the query the user issued and has to be estimated by the auction.

The model assumes that a user will select and visit only service links which are considered relevant for the query. In addition, a user will visit at most one link. The motivation for this assumption is that, in contrast to advertisement in search result pages, a SPARQL query returns the user exactly those links which are of interest for the user, due to the nature of the semantic search. The user can judge the relevancy of the service link by the information embedded in the solution which contains the link. This is different from a WWW search, where users often have to follow a link to discover whether the content provided by the webpage is actually relevant.

If more than one service link is delivered, the user has to decide which of the relevant links to select. Assume that there are n different links, each having a probability of p_{rel} of being relevant to the user. The probability that service link i is the only relevant link is:

$$p_{\text{rel}} \cdot (1 - p_{\text{rel}})^{n-1} \quad (1)$$

In this case, the user will visit this one link because it is to only one relevant.

However, there might be more than one relevant link. In case there are two relevant service links, each relevant link has the same probability of being visited. Hence, the probability the user will select service link i is half the probability that one of the two relevant links is selected:

$$\frac{(p_{\text{rel}})^2 \cdot (1 - p_{\text{rel}})^{n-2}}{2} \quad (2)$$

In total, there are $(n - 1)$ cases having two relevant service links where one of them is link i . In general, there are $\binom{n-1}{k-1}$ cases having k relevant service links where one of them is link i . If we combine all possible cases, we get the formula for the probability $p_{\text{sel}}(n)$ of a link i being selected:

$$p_{\text{sel}}(n) := \sum_{j=1}^n \binom{n-1}{j-1} \cdot \frac{(p_{\text{rel}})^j \cdot (1 - p_{\text{rel}})^{n-j}}{j} \quad (3)$$

COROLLARY 5.2.

$$p_{\text{sel}}(n) := \frac{1 - (1 - p_{\text{rel}})^n}{n} \quad (4)$$

PROOF.

$$\sum_{j=1}^n \binom{n-1}{j-1} \cdot \frac{(p_{\text{rel}})^j \cdot (1 - p_{\text{rel}})^{n-j}}{j}$$

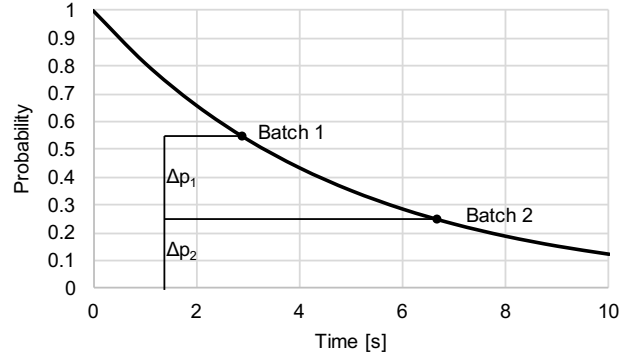


Figure 3: Probability that a user waits for the solutions to be delivered.

$$\begin{aligned} &= \sum_{j=1}^n \binom{n-1}{j-1} \cdot \frac{n}{j} \cdot \frac{(p_{\text{rel}})^j \cdot (1 - p_{\text{rel}})^{n-j}}{n} \\ &= \frac{\sum_{j=1}^n \binom{n-1}{j-1} \cdot (p_{\text{rel}})^j \cdot (1 - p_{\text{rel}})^{n-j}}{n} \\ &= \frac{\left(\sum_{j=0}^n \binom{n-1}{j-1} \cdot (p_{\text{rel}})^j \cdot (1 - p_{\text{rel}})^{n-j} \right) - (1 - p_{\text{rel}})^n}{n} \\ &= \frac{1 - (1 - p_{\text{rel}})^n}{n} \end{aligned} \quad (5)$$

where the last equality holds because the sum in brackets is the distribution formula of the binomial distribution. \square

Introducing delays into the delivery of results adds an additional layer of complexity to our batch link model. Figure 3 shows an example of how the probability can decrease over time. In this example, the first batch is delivered in slightly less than 3 seconds and the second one after almost 7 seconds. The probability that the user waits until the first batch of solutions arrives is around 55%. The probability that the user waits until the second batch of solutions also arrives is around 25%. The probabilities Δp_1 and Δp_2 indicate how likely it is that the user stops waiting after receiving batch 1 and batch 2, respectively, and chooses one of the relevant links received. Let n_j be the number of solutions contained in batch j . We can now define the probability of a link inside a specific batch being selected:

Definition 5.3. Let there be m batches containing n_1, \dots, n_m query solutions, each batch j having a probability of Δp_j of being the last one received by the user. A link i contained in batch $b(i)$ has the following probability p_{sel_i} of being selected by the user:

$$p_{\text{sel}_i} := \sum_{j=b(i)}^m \Delta p_j \cdot p_{\text{sel}}(N_j), \quad (6)$$

where $N_j = \sum_{k=1}^j n_k$

We also define a probability Δp_0 which indicates the likelihood that the user does not wait for the first solution to arrive.

5.2 Ranking Function

To decide which solution should be placed in which batch, they have to be ranked according to the bids placed on the links. There are two well-known ways of ranking the links, either by ranking them by the bids placed on them or by their *revenue*, which is the probability of being selected times the bid. Because of our assumption that the different links have the same probability of being relevant for the user, two different links will have the same probability of being selected for the same position in the ranking. In addition, the probability of being selected is monotonically decreasing with respect to the position. This means that ranking the links by bids results in the same ranking as ranking them by revenue. Hence, we can just order the solutions by the bids they contain to achieve a ranking by revenue. If two links receive the exact same bid, the ranking among the solutions is determined randomly. Note that using a ranking function is not compatible with a user specified order of the solutions and hence, ORDER-BY clauses in SPARQL queries are not supported by our model.

After the solutions are ranked according to the bids on the links they contain, the solutions can be assigned to the different batches. Let n_j the amount of solutions contained in batch j . The first n_1 solutions will be assigned to the first batch, the next n_2 solutions to the second batch, and so on. The parameters n_1, \dots, n_m as well as the delay of the different batches have to be determined in advance. The size of the different batches and the delays determine the probability for the different service links of being selected.

5.3 Weighted VCG

We use the weighted VCG auction [21] to determine the prices the different bidders must pay for each visit by a user. We use this auction mechanism because it is *truthful*, which means that the best strategy for each bidder in this auction is to place a bid which equals their *value* for a visit by the user. The bidder's value for a visit is the *maximal* amount the bidder would be willing to pay for a user's visit. Since the auction is truthful, we will assume that the bidders are bidding their true value and we will use the two expressions *bid* and *value* interchangeably.

The probabilities which are given by the batches, the delays, and the probability p_{rel} are needed to calculate the prices the sponsors have to pay if the user visits a service link. The VCG price π_i for a link i is based on the "harm" the bidder imposes on all other bidders. This harm can be calculated by the difference of the revenue all other bidders would have if there were no bid on link i minus the revenue all other bidders have because there is a bid on link i . The revenue of bidder i must be equal to this difference.

Definition 5.4. Given n links with probabilities $p_{\text{sel}_1}, \dots, p_{\text{sel}_n}$ and bids v_1, \dots, v_n . The weighted VCG price π_i for link i is:

$$\pi_i := \frac{\sum_{\substack{j=1 \\ j \neq i}}^n v_j \cdot (p_{\text{sel}_j}^{\#i} - p_{\text{sel}_j})}{p_{\text{sel}_i}} \quad (7)$$

where $p_{\text{sel}_j}^{\#i}$ denotes the probability of link j being visited if there were no bid on link i .

We show now how this price can be computed more efficiently. For this, we need the following definition:

Definition 5.5. For a link j in batch $b(j)$, we define the 2nd value $v_j^{2\text{nd}}$ as the highest bid of the next batch $b(j) + 1$. If j is in the last batch b_{max} , we define the 2nd value to be 0:

$$v_j^{2\text{nd}} := \begin{cases} \max\{v_k : b(k) = b(j) + 1\}, & \text{for } j < b_{\text{max}} \\ 0, & \text{for } j = b_{\text{max}} \end{cases} \quad (8)$$

Intuitively, the 2nd value $v_j^{2\text{nd}}$ is a lower bound for a bid v for staying in batch $b(i)$. As long as $v > v_j^{2\text{nd}}$, the link can stay in batch $b(i)$. If $v = v_j^{2\text{nd}}$, it is not guaranteed that link i stays in the same batch because the ranking among all links with the same value is random. Finally, if $v < v_j^{2\text{nd}}$, the link ends up in a next higher batch.

Using this definition, we can state the following theorem:

THEOREM 5.6. The weighted VCG price π_i for a link i is given by:

$$\pi_i = \frac{\sum_{j=b(i)}^m v_j^{2\text{nd}} \cdot \Delta p_j \cdot p_{\text{sel}}(N_j)}{p_{\text{sel}_i}} \quad (9)$$

where m is the total number of batches.

PROOF. Equation 7 gives the general form of the weighted VCG price. The first observation is that $p_{\text{sel}_j}^{\#i} = p_{\text{sel}_j}$ for any link j which is positioned before link i , as their position does not change when the bid on link i is removed. For any link j positioned after link i , the probability $p_{\text{sel}_j}^{\#i}$ changes only if link j ends up in a different batch because of the removal of the bid on link i .

There might be multiple service links which have a zero bid on them. Any change in the rankings among links without any bid yields $v_j \cdot (p_{\text{sel}_j}^{\#i} - p_{\text{sel}_j}) = 0$ because $v_j = 0$ and hence, it does not affect the price. Hence, without loss of generality, we can assume that link i gets the last possible position in the ranking when the bid is removed.

If link i receives the last position, the link with the highest value in each batch between $b(i)$ and b_{max} ends up in a next lower batch. This means that $v_j \cdot (p_{\text{sel}_j}^{\#i} - p_{\text{sel}_j})$ can only be non-zero if $v_j = v_{b(j)-1}^{2\text{nd}}$ that is, if v_j is the highest value inside its batch $b(j)$ and hence, the second value of the next higher batch. In these cases, the difference $p_{\text{sel}_j}^{\#i}$ and p_{sel_j} is exactly:

$$\Delta p_j \cdot p_{\text{sel}}(N_j) \quad (10)$$

by Equation 6. Hence:

$$\sum_{\substack{j=1 \\ j \neq i}}^n v_j \cdot (p_{\text{sel}_j}^{\#i} - p_{\text{sel}_j}) = \sum_{j=b(i)}^m v_j^{2\text{nd}} \cdot \Delta p_j \cdot p_{\text{sel}}(N_j) \quad (11)$$

□

6 OPTIMIZING BATCH SIZES AND DELAYS

There are two sets of parameters we must control when setting up the delay auction: the batch sizes n_1, \dots, n_m and their probabilities $\Delta p_1, \dots, \Delta p_m$.

Remember that we have a probability Δp_0 indicating how likely it is that the user does not wait for the first batch of solutions to arrive. The probability Δp_0 is given by the user model and cannot be set by the auction designer. In addition, we set a threshold t_{max}

which indicates the maximal amount of time we are willing to let the user wait until the last solution arrives. The threshold t_{\max} maps to a probability p_{\max} that the user waits for the last batch to arrive.

Let there be n different links. We have three constraints:

$$\sum_{i=1}^m n_i = n, \quad (12)$$

$$\sum_{i=1}^m \Delta p_i = 1 - \Delta p_0, \quad (13)$$

and

$$\Delta p_m \geq p_{\max}. \quad (14)$$

After the probabilities $\Delta p_1, \dots, \Delta p_m$ are determined, the delays of the batches can be chosen to match the desired probabilities.

Note that the parameters n_1, \dots, n_m and $\Delta p_1, \dots, \Delta p_m$ cannot be decided ad hoc when the bids are received. If one would try to optimize the parameters given the bids one would render the auction untruthful despite the use of the weighted VCG mechanism. This is because before the mechanism is applied, the bidders could manipulate the parameters of the auction by manipulating their bids. Hence, the optimizations we discuss in this section have to be done based on what values have to be expected coming up in future auctions.

6.1 Optimizing Revenue

The revenue r generated by our auction mechanism is the sum of the VCG payments of all sponsors involved in a specific query answer. As we showed in Section 5.3, the payments can be calculated according to Equation 7. The total revenue generated by the auction is:

$$\begin{aligned} r &= \sum_{i=1}^n \pi_i \cdot p_{\text{sel}_i} \\ &= \sum_{i=1}^n \sum_{j=b(i)}^n v_i^{2\text{nd}} \cdot \Delta p_j \cdot p_{\text{sel}(N_j)} \end{aligned} \quad (15)$$

If we concentrate first on the best choice for the probabilities $\Delta p_1, \dots, \Delta p_m$, we see that the revenue is a linear function with respect to those probabilities. This means that the revenue is maximized when one of the probabilities is set to 1 and all others to 0. Let Δp_{\max} be the probability which maximizes the revenue when set to 1. If we also consider the constraint in Equations 13 and 14, the revenue is maximized when $\Delta p_{\max} = 1 - \Delta p_0$ and $\Delta p_m = p_{\max}$ or, if $\max = m$, $\Delta p_{\max} = 1 - \Delta p_0$, respectively. All other probabilities are set to 0. Setting $\Delta p_j = 0$ means that batch j has the same delay as the next batch $j + 1$ and practically, they become one batch.

Optimizing the batch sizes is not as straightforward as optimizing the probabilities. But, since there are at most two batches remaining after optimizing the probabilities, the problem becomes easier. If $\max = m$ the revenue is 0, as there is only one batch and hence, the second value is 0. If there are two batches left we redefine the two remaining probabilities as Δp_1 and Δp_2 and denote with n_1 and n_2 the number of solutions in the two batches. We will denote the

revenue when having only two batches as r^* :

$$\begin{aligned} r^* &= \sum_{j=1}^{n_1} v_1^{2\text{nd}} \cdot \Delta p_1 \cdot \frac{1 - (1 - p_{\text{rel}})^{n_1}}{n_1} \\ &= v_1^{2\text{nd}} \cdot (1 - \Delta p_0 - p_{\max}) \cdot (1 - (1 - p_{\text{rel}})^{n_1}) \end{aligned} \quad (16)$$

Note that the value $v_1^{2\text{nd}}$ depends on the choice of n_1 . The only way to optimize r^* is to iterate through all possible values of n_1 . Fortunately, the possible values for n_1 are discrete and bounded by n and hence, the problem can be solved in $O(n)$ time.

6.2 Optimizing Social Welfare

The social welfare s generated by our auction mechanism is the sum of the values of all links times the probability that the link is visited. Since the auction is truthful, we can assume that the bids equal the bidders actual value for being selected. Hence, the social welfare is given by the following formula:

$$\begin{aligned} s &= \sum_{i=1}^n v_i \cdot p_{\text{sel}_i} \\ &= \sum_{i=1}^n v_i \cdot \left(\sum_{j=b(i)}^m \Delta p_j \cdot p_{\text{sel}(N_j)} \right) \end{aligned} \quad (17)$$

The same argument which applies to maximizing the revenue also applies here: the social welfare is a linear function with respect to those probabilities, which is maximized if we set $\Delta p_{\max} = 1 - \Delta p_0 - p_{\max}$ and $\Delta p_m = p_{\max}$, or $\Delta p_{\max} = 1 - \Delta p_0$, respectively, and all others to 0. Again, to optimize social welfare the auctioneer has to split the solutions into at most two batches. Similar to r^* , we denote the social welfare when having two batches as s^* :

$$\begin{aligned} s^* &= \left(\sum_{i=1}^{n_1} v_i \right) \cdot (1 - \Delta p_0 - p_{\max}) \cdot \frac{1 - (1 - p_{\text{rel}})^{n_1}}{n_1} + \\ &\quad \left(\sum_{i=1}^{n_2} v_i \right) \cdot p_{\max} \cdot \frac{1 - (1 - p_{\text{rel}})^{n_2}}{n_2} \end{aligned} \quad (18)$$

As with r^* , s^* can be maximized by iterating through all possible values for n_1 . There are two properties of the maximal value for s^* worth noting. First, the parameter n_1 which maximizes s^* does not maximize r^* in general. This is easy to see when comparing Equation 16 with Equation 18: r^* depends on the value $v_1^{2\text{nd}}$ whereas s^* depends on all values. Second, choosing $n_1 = n$ does not maximize s^* in general. This is because the more solutions are included in the first batch, the more likely it is that a solution having a low value will be chosen.

6.3 Optimizing User Experience

The user's experience is an important aspect of our auction we have not discussed so far. Even if the data the auction is offering to the user is for free, a user might switch to a competitor offering a similar service if the competitor can offer a better user experience.

We have already seen that in both cases, optimizing revenue and optimizing social welfare, the optimal choice of parameters results in two batches, one delivered as fast as possible and the second one delivered at time t_{\max} . Delaying a potentially big chunk of the available solutions for this maximal delay t_{\max} might damp

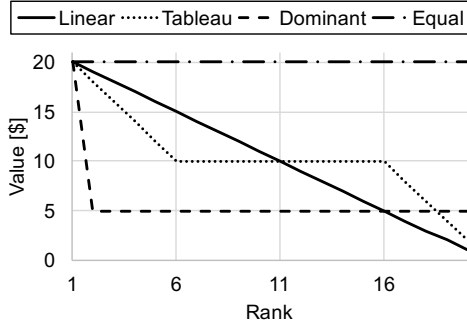


Figure 4: Value distribution for four different scenarios.

the user’s experience, however. Of course, the user would prefer to receive all data in the first batch, which is delivered immediately. However, as we have seen in the previous section, delivering all data at once results in zero revenue and possibly suboptimal social welfare. Fortunately, it is possible delaying some solutions without impacting the user experience too much. The main argument is that a user will need some time to consume the solutions contained in a batch. If the delay of the next batch is not larger than the time the user needs to consume the current batch, the user will be able to seamlessly consume all the solution from the different batches.

Another aspect which can damp the user’s experience is the fact that the delivery of the data is biased in the sense that highly sponsored solutions are always delivered first. In Section 7, we discuss an extension of our model which reduces this bias.

6.4 Simulation

In this section, we illustrate how revenue and social welfare behave when there are only two batches. Figure 4 shows four different value distributions which we labeled with Linear, Tableau, Dominant, and Equal, respectively. For our simulation, we set $\Delta p_0 = 0.05$, $p_{rel} = 0.6$, and $p_{max} = 0.1$.

Figure 5 shows the revenue for different choices of n_1 . The revenue always starts at \$0 for $n_1 = 0$ and closes with \$0 for $n_1 = 20$. This is no surprise, as in both cases all solutions are only assigned to a single batch and hence, the second price is \$0. There are two remarkable observations about the revenue, however. First, for the Tableau distribution, the revenue stays the same for $n_1 = 5$ up to $n_1 = 15$. This happens because the second value stays constant within this range, as the bids are all the same. Second, the Dominant and Equal distributions both have their maximum at $n_1 = 19$. The reason for this is that the highest bid within any distribution does not influence the revenue.

Figure 6 shows the social welfare for different choices of n_1 . The social welfare is always positive, no matter the choice of n_1 . For the Equal distribution the social welfare is maximized for $n_1 = 20$, but does not change much after n_1 is beyond seven. The values for n_1 which maximize the social welfare for the Linear and Tableau distributions, respectively, are quite close to the ones which maximize their respective revenue. The Dominant distribution has its maximum for the social welfare for $n_1 = 1$. Increasing n_1 further increases the chance that a link with a lower value is chosen by

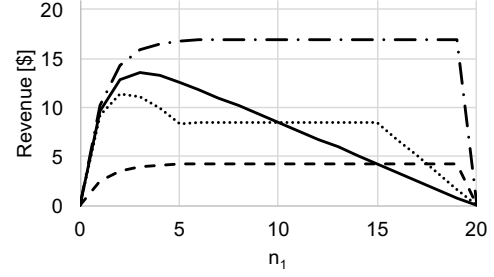


Figure 5: Revenue for different values for n_1 for the different value distributions from Figure 4.

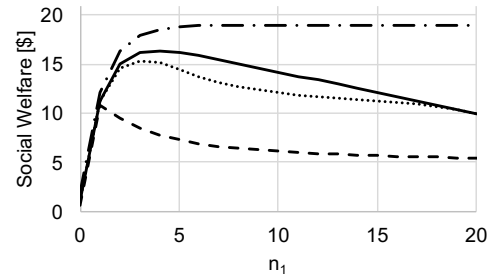


Figure 6: Social welfare for different values for n_1 for the different value distributions from Figure 4.

the user and hence, decreases social welfare. The Dominant distribution illustrates how far away the values for n_1 can be which maximize social welfare and revenue, respectively.

7 EXTENSION

To mitigate the bias our auction introduces by delivering the highest sponsored data first, we propose an extension of our original model. The idea of this extension is to auction off only part of the available positions in the batches and assign the rest of them randomly. The result of this extension is similar to what users experience when using search engines: Part of the results is sponsored content and ranked according to some bids, the other part of the results is unbiased and only depends on what the user is searching for.

The first step in our extension is to designate some slots in each batch which should be reserved for sponsored solutions. We will denote with n_i^{spons} and n_i^{rand} how many of the n_i slots in batch i are reserved for sponsored solutions or are assigned randomly, respectively. The next step is to assign solutions to the random slots. We denote with n^{spons} and n^{rand} the total number of sponsored slots and random slots, respectively, and with n the total number of slots, which equals the total number of solutions for the query. Note that once we have assigned all random slots, there are $n - n^{rand}$ solutions which did not receive any slot, yet. These solutions are randomly arranged in a waiting queue, where the first link in the queue will be the first one to occupy a newly open random slot.

The next step is to assign solutions to the sponsored slots. We start with the assignment of the n_1^{spons} slots in the first batch. For

this, we select the n_1^{spons} solutions with the highest bid from all those solutions which are not already assigned to the first batch by the random assignment. This means that some of the solutions get upgraded to the first batch. An upgraded solution can be either one of the solutions in the waiting queue or one of the solution which already had a randomly assigned slot. In the latter case, the upgrading creates a new open position for a randomly assigned slot, which is assigned to first link in the waiting queue. This procedure is applied sequentially to the sponsored slots n_2^{spons} to n_m^{spons} . For every batch, we only consider those solutions which are currently in a higher (and later) batch or the waiting queue.

Once we have the new ranking of the solutions, we can again calculate the weighted VCG prices according to Definition 5.4. Note that the Theorem 5.6 does not anymore apply to this new setting.

We conclude our extension with an example:

Example 7.1. Assume that there are four links with bids of \$0, \$10, \$20, and \$30. The left part in Figure 7 shows a random assignment of the bids \$0 and \$30. The other two bids, \$10 and \$20, are assigned to the waiting queue. The bid \$20 takes precedence over \$10 inside the queue. The right part in Figure 7 shows the final assignment of the bids. The arrow labeled (A) indicates the first step of the assignment of the sponsored slots: bid \$30 gets to the first sponsored slot, because it is the highest bid of those three bids, \$30, \$20, and \$10, which are not yet in batch 1. Next (B), the bid \$20 gets the random slot previously occupied by \$30, because bid \$20 has the priority in the queue. Finally (C), bid \$10 gets the last remaining sponsored slot because it is the highest (and only) bid which did not yet get a slot in batch 1 or 2. The weighted VCG price is \$0 for all solutions which got a randomly assigned slot, eventually, because changing their bid to \$0 does not influence any other assignments of slots. This means that the bidders with bids \$0 and \$20 do not pay if the user visits their link. The bidder with bid \$10 also gets a price of \$0. This, because the bidder would have gotten in the second batch even with a bid of \$0. Finally, the weighted VCG price for the bidder with \$30 is the value all other bidders would have if bid \$30 would have been \$0, minus the value all other bidders have for the current assignment.

Assume that $p_{\text{rel}} = 0.8$, $\Delta p_1 = 0.5$, and $\Delta p_2 = 0.4$. Hence, the probabilities of being selected for the two first slots are 0.20096 each, and for the second two slots 0.04096 each. If bid \$30 would be zero, the first sponsored slot would be assigned to bid \$20 and the first random slot would still belong to the bidder with the original bid \$0. The second sponsored slot would be assigned to bid \$10. The second random slot would be assigned to the bidder with the original bid of \$30. The total value of this assignment is \$4.4288. The value of all other bidders for the original assignment is \$1.2288. With a probability of 0.20096 of being selected, the weighted VCG price for the bidder with bid \$30 is \$15.92, which has to be paid if the user visits the link.

8 LIMITATIONS AND CONCLUSIONS

We introduced a new concept of a delayed-answer auction to finance free data in the WoD. As we have seen, the choice of parameters can influence the generated revenue, generated social welfare, and the user experience. In general, it is not possible to find parameters which maximizes all three of them. Hence, it is the choice of the

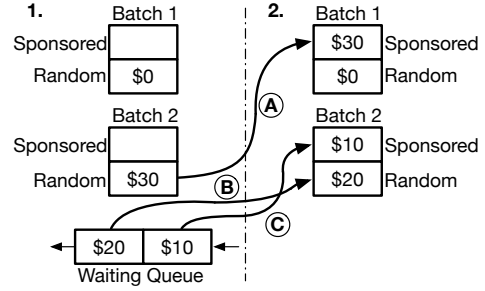


Figure 7: Example: four links get assigned to different slots.

auction designer to find a suitable trade-off between revenue, social welfare, and user experience. We also discussed an extension which reduces the bias we introduce by prioritizing solutions containing links with high bids.

Our new auction model is not restricted to the use case of the WoD. It can be applied to any setting where multiple bidders can occupy the same slot and different slots have a decreasing probability of being selected. An additional restriction is that we assume that all solutions have the same relevance to the user. As we discussed, this assumption is reasonable in the WoD setting. In other settings, this assumption might not apply. In this case, the auction might become more complicated because the probabilities of being selected might not be separable anymore and hence, the weighted VCG auction is not applicable.

One additional assumption we made is that there is only one sponsor per solution. While this assumption might be true in most cases, it is possible that a user might issue a query which contains multiple sponsored links per solution. In such situations, the bid per solution can be defined as the sum of the bids for each link contained in the solution. With this definition, we could still apply our auction model, but we would have to find a way to distribute the weighted VCG payment among the bidders which placed a bid on the same solution. If one decides to distribute the payments proportional to the bids placed on the links, the auction would not anymore be truthful, however.

What is left for future work is the distribution of the generated revenue among the data providers. The revenue can be used to finance those providers which proved to be important for answering queries or subsidize those providers which struggle the most to keep their services running.

With our concept of a delayed-answer auction, we provided a first sponsored auction model for the Web of Data. Whatever the limitations of our concept are, it represents a first model for a financially sustainable and free Web of Data.

ACKNOWLEDGMENTS

This work was partially supported by the Swiss National Science Foundation under grant #153598 (<http://p3.snf.ch/project-153598>).

REFERENCES

- [1] Gagan Aggarwal, Jon Feldman, and Shanmugavelayutham Muthukrishnan. 2007. Bidding to the Top: VCG and Equilibria of Position-Based Auctions. In *Approximation and Online Algorithms. WAOA 2006*.
- [2] Gagan Aggarwal, Jon Feldman, Shanmugavelayutham Muthukrishnan, and Martin Pál. 2008. Sponsored Search Auctions with Markovian Users. In *WINE '08*. 621–628.
- [3] Gagan Aggarwal, Ashish Goel, and Rajeev Motwani. 2006. Truthful auctions for pricing search keywords. In *EC '06 Proceedings of the 7th ACM conference on Electronic commerce*. 1–7.
- [4] Darko Anicic, Paul Fodor, Sebastian Rudolph, and Nenad Stojanovic. 2011. EP-SPARQL: a unified language for event processing and stream reasoning. In *WWW. ACM*, 635–644.
- [5] Davide Francesco Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, and Michael Grossniklaus. 2010. C-SPARQL: A Continuous Query Language for RDF Data Streams. *International Journal of Semantic Computing* 04, 01 (2010), 3–25.
- [6] Jean-Paul Calbimonte, Oscar Corcho, and Alasdair J. G. Gray. 2010. Enabling Ontology-based Access to Streaming Data Sources. In *Proceedings of the International Semantic Web Conference ISWC '10*. 96–111.
- [7] Edward H. Clarke. 1971. Multipart Pricing of Public Goods. *Public Choice* 2 (1971), 19–33.
- [8] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In *WSDM '08 Proceedings of the 2008 International Conference on Web Search and Data Mining*. 87–94.
- [9] Daniele Dell'Aglio, Emanuele Della Valle, Frank van Harmelen, and Abraham Bernstein. 2017. Stream reasoning: A survey and outlook. *Data Science* 1, 1–2 (2017), 59–83.
- [10] Benjamin Edelman and Michael Ostrovsky. 2007. Strategic Bidder Behavior in Sponsored Search Auctions. In *Decision Support Systems*, Vol. 43. 192–198.
- [11] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. 2007. Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars. *The American Economic Review* 97, 1 (2007), 242–259.
- [12] Lee Feigenbaum, Gregory Todd Williams, Kendall Grant Clark, and Elias Torres. 2013. SPARQL 1.1 Protocol. <https://www.w3.org/TR/sparql11-protocol/>. (March 2013).
- [13] Ruediger Glott, Philipp Schmidt, and Rishab Ghosh. 2010. *Wikipedia survey – overview of results*. Technical Report. United Nations University MERIT.
- [14] Theodore Groves. 1973. Incentives in Teams. *Econometrica* 41(4) (1973), 617–631.
- [15] Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi-Min Wang, and Christos Faloutsos. 2009. Click Chain Model in Web Search. In *WWW '09 Proceedings of the 18th international conference on World wide web*. 11–20.
- [16] James Hamilton. 2009. The Cost of Latency. *Perspectives* (October 31 2009). <http://perspectives.mvdirona.com/2009/10/the-cost-of-latency/>
- [17] Steve Harris and Andy Seaborne. 2013. SPARQL 1.1 Query Language. <https://www.w3.org/TR/sparql11-query/>. (March 2013).
- [18] Ron Kohavi, Roger Longbotham, Dan Sommerfield, and Randal M. Henne. 2009. Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery* 18, 1 (2009), 140–181.
- [19] Danh Le-Phuoc, Minh Dao-Tran, Josiane Xavier Parreira, and Manfred Hauswirth. 2011. A Native and Adaptive Approach for Unified Processing of Linked Streams and Linked Data. In *Proceedings of the International Semantic Web Conference ISWC '11*. 370–388.
- [20] Steve Lohr. 2012. For impatient web users, an eye blink is just too long to wait. *New York Times* (February 29 2012). <http://www.nytimes.com/2012/03/01/technology/impatient-web-users-flee-slow-loading-sites.html>
- [21] Noam Nisan and Amir Ronen. 2007. Computationally Feasible VCG Mechanisms. *Journal of Artificial Intelligence Research* 29 (2007), 19–47.
- [22] Özgür Lütü Özçep, Ralf Möller, and Christian Neuenstadt. 2014. A Stream-Temporal Query Language for Ontology Based Data Access. In *KI (Lecture Notes in Computer Science)*, Vol. 8736. Springer, 183–194.
- [23] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting Clicks: Estimating the Click-Through Rate for New Ads. In *WWW '07 Proceedings of the 16th international conference on World Wide Web*. 521–530.
- [24] Mikko Rinne, Esko Nuutila, and Seppo Törmä. 2012. INSTANS: high-performance event processing with standard RDF and SPARQL. In *Proceedings of the ISWC 2012 Posters & Demonstrations Track*.
- [25] Hal R. Varian and Christopher Harris. 2014. The VCG Auction in Theory and Practice. *American Economic Review* 104, 5 (2014), 442–45.
- [26] William Vickrey. 1961. Counterspeculation, Auctions, and Competitive Sealed Tenders. *The Journal of Finance* 16(1) (1961), 8–37.
- [27] Christopher A. Wilkens, Ruggiero Cavallo, and Rad Niazadeh. 2017. GSP – The Cinderella of Mechanism Design. In *WWW '17 Proceedings of the 26th International Conference on World Wide Web*. 25–32.
- [28] Zeyuan Allen Zhu, Weizhu Chen, Tom Minka, Chenguang Zhu, and Zheng Chen. 2010. A novel click model and its applications to online advertising. In *WSDM '10 Proceedings of the third ACM international conference on Web search and data mining*. 321–330.